Untangling Web Query

How metadata can reduce query and report complexity

As printed in the September 2009 edition of the IBM Systems Magazine

Written by Gene Cobb – cobbg@us.ibm.com

What is Metadata?

Since DB2 Web Query was announced in March 2007, I've delivered countless webcasts, presentations, workshops and labs to educate the IBM i community about this new technology. I'm constantly asked about Web Query's metadata requirement. Many i customers are simply not accustomed to working with a product that requires metadata and consequently view it as an extra layer of complexity they must create and maintain.

In this article, I'll provide answers to the questions I've been hearing about what metadata is, why i customers need it and what happens if they change the structure of the underlying files. Metadata is a good thing. It can actually reduce the complexity of your query and reporting environment and make life easier for your report developers.

Benefits of Metadata

The metadata layer's primary objective is to improve DB2 Web Query report developers' productivity by providing an abstraction layer and burying the database complexity. Report developers can be more productive if you keep the data model simple. A simple and intuitive data model also means you can extend the report developer community because it doesn't require intimate knowledge of the database. By empowering more of your users, you can reduce their dependence on IT and the number of backlogged reporting requests.

But before this can happen, you'll need to build up this abstraction layer, which requires the DB2 Web Query Developer Workbench tool. This Windows* PC-based tool offers several powerful components, one of which is the Synonym Editor. The Synonym Editor contains graphical interfaces that let you:

- Define database joins
- Build virtual columns to centralize business logic
- Standardize column formats
- Convert and standardize date fields
- Create filters
- Create business views
- Define online analytical processing (OLAP) dimensions

I'll share some details on each of these features in the remainder of this article. If you want more hands-on tips, read the Web-exclusive article, "Get a Handle on Metadata" (see article list), which includes detailed instructions and explanatory figures.

Define Database Joins

If you've developed reports with Query/400, you're likely aware that if the query requires more than one file, you must define your joins in each query definition. This may seem perfectly acceptable, but wouldn't it be nice if you could just define your joins in one place and base all of your reports on that single, centralized definition? That's exactly what you can build in the metadata layer for DB2 Web Query reports.

Using the Synonym Editor in Developer Workbench, you can add new join segments from existing synonyms or files that simply need to be brought into the cluster (and don't require standalone synonyms). Then you must define the join fields and the Synonym Editor will group the columns under their respective segments and provide an interface from which the segments can be expanded and collapsed.

The Synonym Editor also provides a graphical representation of the model in the Modeling View tab. You can double-click on any of the segments to show the column information and sample data for that particular file, as shown in Figure 1.

Because the join segments have been defined in the synonym, you won't need to define them again in the report, which can greatly enhance developer productivity when creating multiple reports over the same data source.

Build Virtual Columns to Centralize Business Logic

If yours is like many companies, you have fields in your database files that store values with specific, perhaps hidden, meanings and contain conditional rules. For example:

- The second character of column ORDXYZ is a code that represents the order status.

- If column ORDABC = “S” then the value for column ORDCBA must be multiplied by -1.
- A unit price calculation (PRICE * UNITS) is based on another field to correctly calculate the quantity UNITS (“M” = 100, “X” = 1,000, etc.).
- If column CFLAG = “Y” then the next row with the same order number key is a continuation of the current row.

I could list dozens of examples, but the point is many databases contain business logic specific to the company and buried in hundreds of RPG and COBOL programs. It’s a more efficient, application-independent and data-centric approach to move this logic either down the database itself (using SQL views, stored procedures, triggers and referential integrity) or into the DB2 Web Query metadata layer. Centralizing the logic also means you’ll have one version of the truth. This helps eliminate errors and enhance productivity because developers without the business knowledge don’t need to worry about how to define gross profit margin in every report.

These calculated and conditional fields can be abstracted in DB2 Web Query by creating virtual columns in the synonym. For example, let’s suppose there are two joined tables, ORDERS and STORES. The STORES table contains a field named STORESUBCODES, which is a 10-character string with each character representing something different for that particular store. The fourth character of this field is used to calculate order discounts. The valid values are 0-9 and each value represents a specific discount rate. A value of zero means the store receives no discount, and value of 9 means the store is granted a 20-percent discount. Any value between those two equates to a specific discount somewhere in that range.

To develop a series report that calculates measures such as gross profit for each store, you’d have to know and understand this business rule and apply it to every report. It’d be much easier to perform this in the metadata so it only must be defined once. Two virtual columns in the metadata could store and apply the business logic for you.

Standardize Column Formats

If your reports include numeric currency columns such as revenue and gross profit, you’ll likely require some standard formatting of these values. Formatting (such as the inclusion of commas as thousands separators, currency symbols and suppression of leading zeroes) enhances the readability and comprehension of the report data. Once again, you can define this in the metadata so you don’t have to repeat the formatting definitions across hundreds of reports. The Synonym Editor provides an interface to easily select these formatting specifications.

Convert and Standardize Date Fields

I’ve worked with many IBM i shops that store their date values in a data type that isn’t a date or timestamp, often because the database was created before these data types were available. Consequently, it’s common to find databases with packed decimal and character fields that actually store date values. An example of these legacy date fields is a field defined as packed decimal (8,0), which contains the value 06252009 to represent the date June 25, 2009.

Because they aren't true date fields, they can be problematic for DB2 Web Query: It simply doesn't know these values represent dates. To Web Query, these fields are just regular packed decimal fields. While this may be fine for storing the date values, it's probably not how you want dates displayed on your reports. DB2 Web Query can convert the legacy date field using a new virtual column that's a true date field (see Figure 2).

Create Filters

Suppose you want to create a series of revenue and gross profits reports for all of the countries in Europe. This sounds easy enough, but what if your database didn't include a column for the continent? Each report would have to specify selection criteria for each of the (roughly 49) countries in Europe, which would impose some serious maintenance challenges for the report developer. Additionally, if the developer wanted to remove one of these countries from the reports, every report would have to be opened and updated.

DB2 Web Query filters can reduce this complexity by allowing the metadata administrator to define all of the European countries in one place—in the synonym. Filters are selection criteria (translated to SQL WHERE statements) that are stored in the synonym and provide the report developer with quick and easy access to the predefined selection criteria that limit the data included in a report. For example, the filter can contain each of the countries in Europe.

Then for any report based on that synonym, the new filter appears in the list of available fields, and you can simply drag it into the screening conditions pane under the Selection criteria tab and run the report to show only orders for the European customers.

Create Business Views

If you have anything but the simplest of data models and are joining multiple tables together, a common problem for many report developers is wading through the model and determining what fields are meaningful and should be included in the report. The data source may have dozens, if not hundreds, of columns to choose from, many of which may have duplicate, ambiguous and cryptic names with no descriptions. With so many columns, developers may have to navigate through multiple pages to locate the correct column. All of this takes away from productivity and increases the chance of errors in the report-development process.

Business views can greatly simplify the report-development process by letting the metadata administrator select and organize relevant columns. This organized subset is stored in a business view, a separate synonym that can be selected as the report's data source.

When a report is based on a Business View synonym, the report developer no longer has a list of many available fields to choose from. Instead, a smaller set is presented. Organizing subsets of the columns and assigning meaningful names and descriptions to them will go a long way in reducing confusion and complexity. Business views can be thought of as an additional abstraction layer with the consistent metadata theme—make it as easy as possible for your report authors to develop the reports that meet your company's requirements.

Define OLAP Dimensions

OLAP is a type of business-intelligence technology that lets users query and analyze data set up in a multidimensional format. Users can interact with the data to perform quick and sophisticated analysis (e.g., drill down through the dimensions of a hierarchy, slice and dice the data in various ways to show a report from a specific perspective and dynamically drag new columns into the report). Because it gives end users so many ways to analyze the data, a single DB2 Web Query OLAP report can often replace dozens of Query/400 reports. Typically this technology is associated with proprietary cubed views, a copy of the data that's built in a multidimensional (cubed) format. With DB2 Web Query, creating a new copy of the data is unnecessary because the OLAP dimension can be defined in—you guessed it—the metadata.

From the Modeling View tab of the Synonym Editor, you can define the dimension and drag columns from the synonym into each of those dimensions. Once your OLAP dimensions are set up, you can transform very simple reports into high-powered, feature-rich OLAP reports (see Figure 3).

Refreshing Metadata

Many customers are concerned about what happens when the metadata gets out of synch with the underlying data source. For example, if the database administrator alters a table and either adds or removes a column, do existing reports still run successfully?

The answer is, it depends. The synonym may need to be refreshed to reflect the change. If none of your reports references a column deleted from the underlying file, a synonym refresh isn't required. The same applies if you don't want to include the new column in any existing or new reports. However, if the changes impact your reports, you'll need to synchronize the synonym with the altered file.

The DB2 Web Query Developer Workbench offers an option from the right-click menu to refresh the synonym. This parses the synonym and compares its contents with the structure of the underlying data source. New columns are added to the synonym and deleted columns are removed. Any custom changes (e.g., joins, virtual columns, filters, etc.) made to the synonym are preserved.

New Feature: Segmentation and Security

A recent enhancement to DB2 Web Query is the capability to segment and secure your metadata. Previously, all synonyms were created in the common application folder BASEAPP, which meant they were displayable and selectable from all report-development interfaces (across all domains). You couldn’t filter the list of synonyms for specific users or groups of users. This was problematic for several reasons:

- All report developers could see “sensitive” synonyms. Although they couldn’t see the data of the underlying tables, the ability to see the names and existence of sensitive synonyms was still a concern.
- The list of synonyms to use when creating a new report could be extensive, making finding the desired synonym difficult.
- Promoting reports and referenced synonyms through the development cycle was tedious because the reports had to be changed (at each step) to reference specific and uniquely named synonyms in the BASEAPP folder.

IBM has addressed these concerns with the new capability to create custom application folders and associate them with specific domains. You can now set up the application folder (and the synonyms within it) as an extension of the domain. Because domains are secured using IBM i group profiles, you can segment and secure your synonyms.

A Good Thing

I hope I’ve convinced you that metadata and a more data-centric reporting environment is, indeed, a good thing. The DB2 Web Query metadata layer is all about abstraction, centralization and simplification. It requires an investment to build up, but the plentiful features in DB2 Developer Workbench provide the tools to complete the job. And by doing so, you’ll pave the way for enhanced efficiency and report accuracy. You’ll have more productive developers and potentially more of them because they no longer need a deep understanding of the data model.

For more information on DB2 Web Query, I suggest the IBM Redbooks* publication “Getting Started with DB2 Web Query for System i” (SG247241, www.redbooks.ibm.com/abstracts/sg247214.html). Don’t forget I’ve provided some how-to details for each of these Synonym Editor features online.

What is Metadata?

Side box to main article

Metadata is simply data about data. Whenever you issue the Display File Description (DSPFD) or Display File Field Description (DSPFFD) commands, what’s generated and displayed on your screen is, in fact, metadata—information such as record lengths, record formats, field names, data types, field attributes and field lengths. DB2 for i also maintains system catalogs, which are files that store information about each object in your database. They're effectively a materialized metadata repository that's kept up-to-date and can be queried to collect a wide variety of information about your database.

In much the same way, DB2 Web Query metadata is a materialized repository that contains information about your database files. Before you can create a report or graph in DB2 Web Query you must create metadata (also referred to as synonyms) over the data sources. You can create a synonym over such database objects as tables or physical files, SQL views, DDS logical files, stored procedures and materialized query tables.